DATA-
BASE

114

SERVER

116

100

DEVICE
INTERFACE

104

COMPONENTS

106

TESTER

102

COMPUTER
SYSTEM

108

PROCESSOR

110

MEMORY

112

FIG. 1

FIG.2

300

START

INITIALIZE — 302

READ APPLICATION CONFIGURATION INFORMATION — 304

CONFIGURE SYSTEM — 306

GET TOOL IDENTIFICATION — 308

END

FIG. 3

START

COPY TESTER DATA FILE TO TOOL INPUT DIRECTORY — 402

READ LOGISTICS FROM TDF — 404

CREATE LOGISTICS INSTANCE WITH TOOL INFORMATION — 406

LOOK UP TEST PROGRAM IDENTIFICATION IN MASTER DATABASE — 408

FOUND ? — 410

Y

N

CREATE TEST PROGRAM IDENTIFICATION AND ASSOCIATE WITH TOOL IDENTIFICATION — 412

GET WAFER TO ANALYZE INFORMATION — 414

A

FIG. 4A

A

WAIT FOR NEXT WAFER   418

ANALYZE
CURRENT
WAFER
?   N

416

Y

GET SECTION GROUP   420

LOAD CONTROL LIMITS AND ENABLE   422
FLAGS FOR TEST PROGRAM FROM DATABASE

GET STATISTIC TYPES FROM SECTION GROUP   423

MATCH SIGNATURE ANALYSIS ALGORITHMS   424
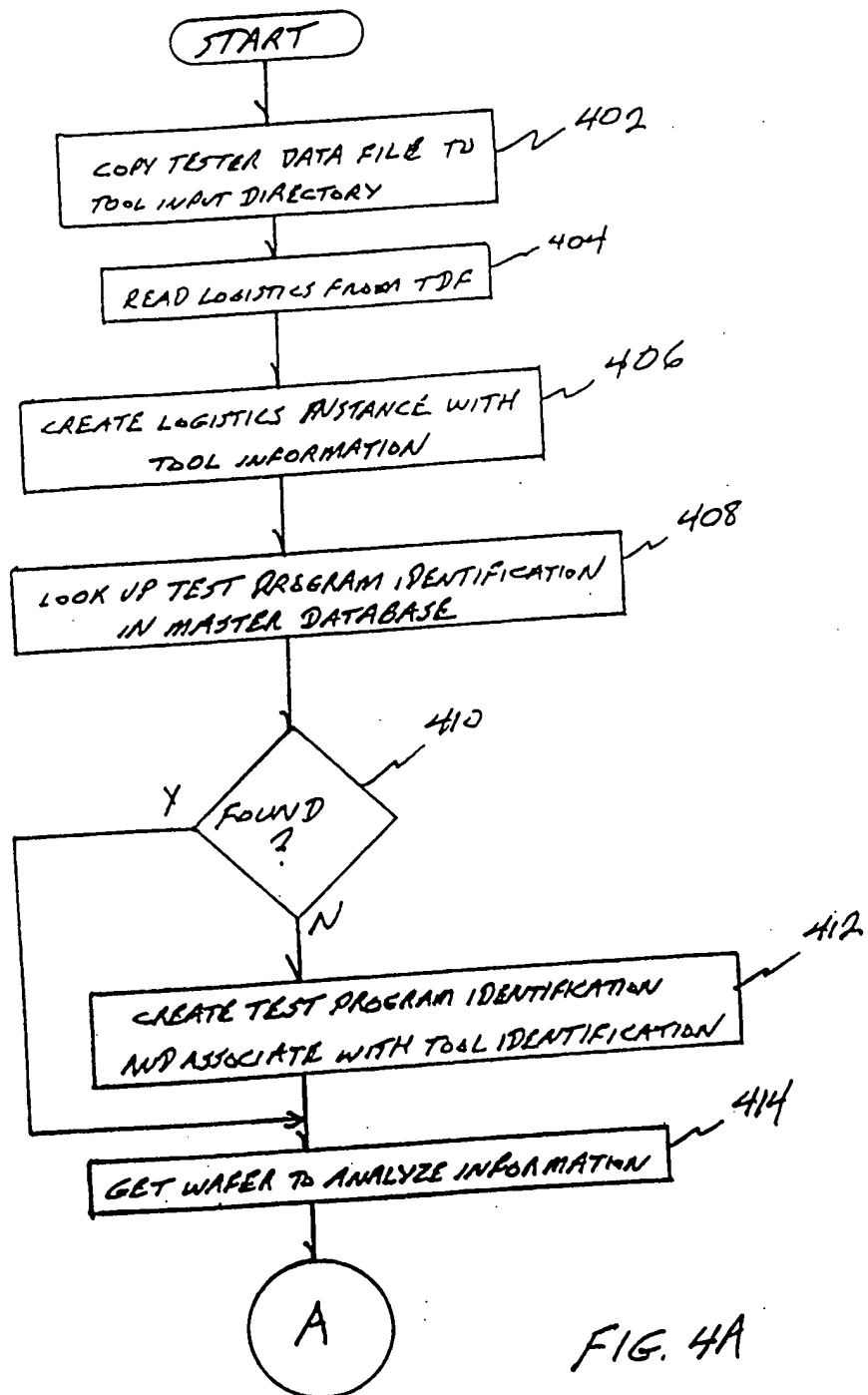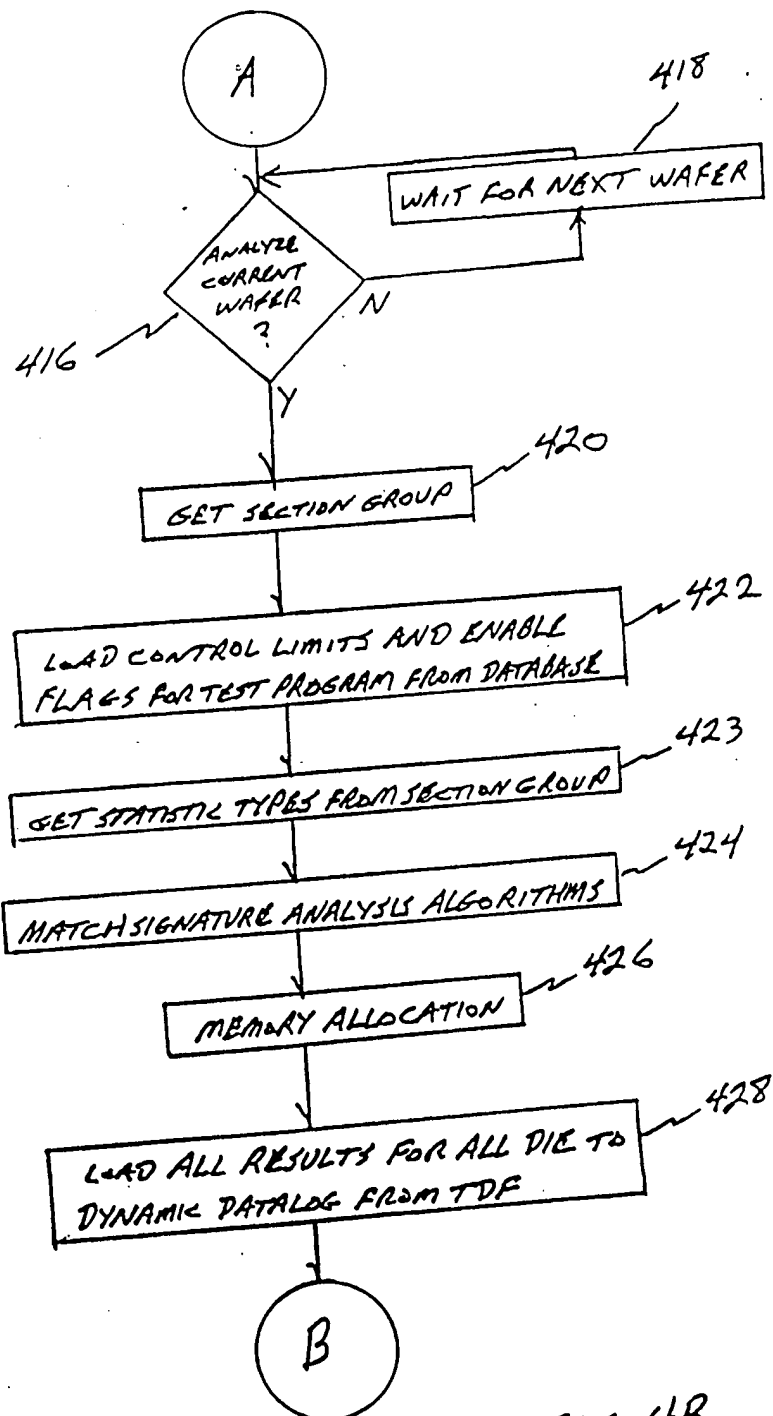
MEMORY ALLOCATION   426

LOAD ALL RESULTS FOR ALL DIE TO   428
DYNAMIC DATALOG FROM TDF

B

FIG. 4B

FIG. 4C

FIG. 5

START

GET POINT COUNT & DIE COUNT `602`

APPLY ALGORITHM TO SECTION DATA POINTS `604`

STORE TREND RESULTS AND SIGNATURE RESULTS `606`

NEXT ALGORITHM ? `608`

NEXT TEST ? `610`

NEXT SECTION ? `612`

A

FIG. 6A

A

RECORD ERRORLOG IN DATABASE — 614

RECORD TREND RESULTS IN DATABASE — 616

RECORD SIGNATURE RESULTS IN DATABASE — 618

END

FIG. 6B

START

READ SAMPLING RANGE — 702

READ OUTLIER INFORMATION — 704

MERGE SAMPLING RANGE AND OUTLIER INFORMATION — 706

STORE MERGED INFORMATION IN OUTPUT INFORMATION — 708

CREATE OUTPUT REPORT — 710

SEND OUTPUT REPORT TO DESTINATION — 712

END
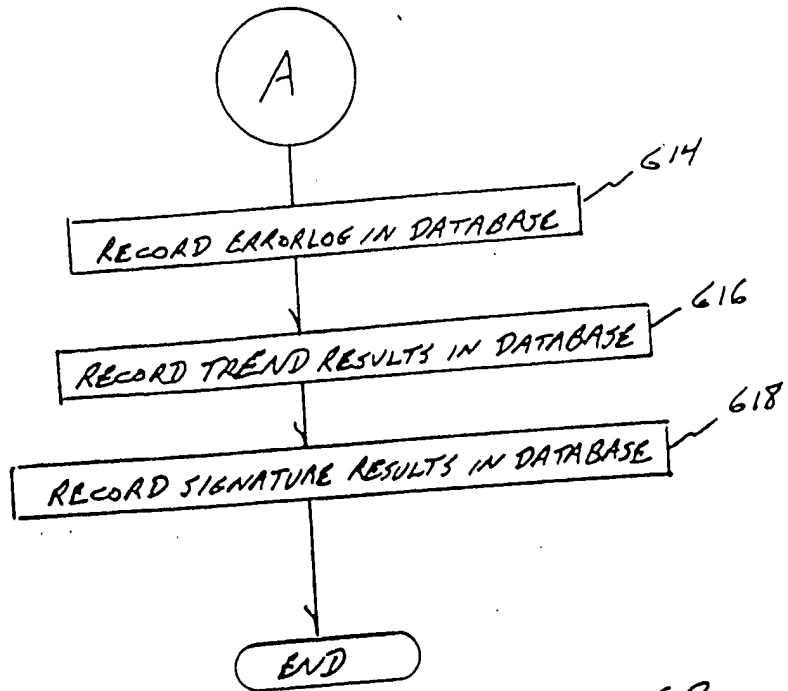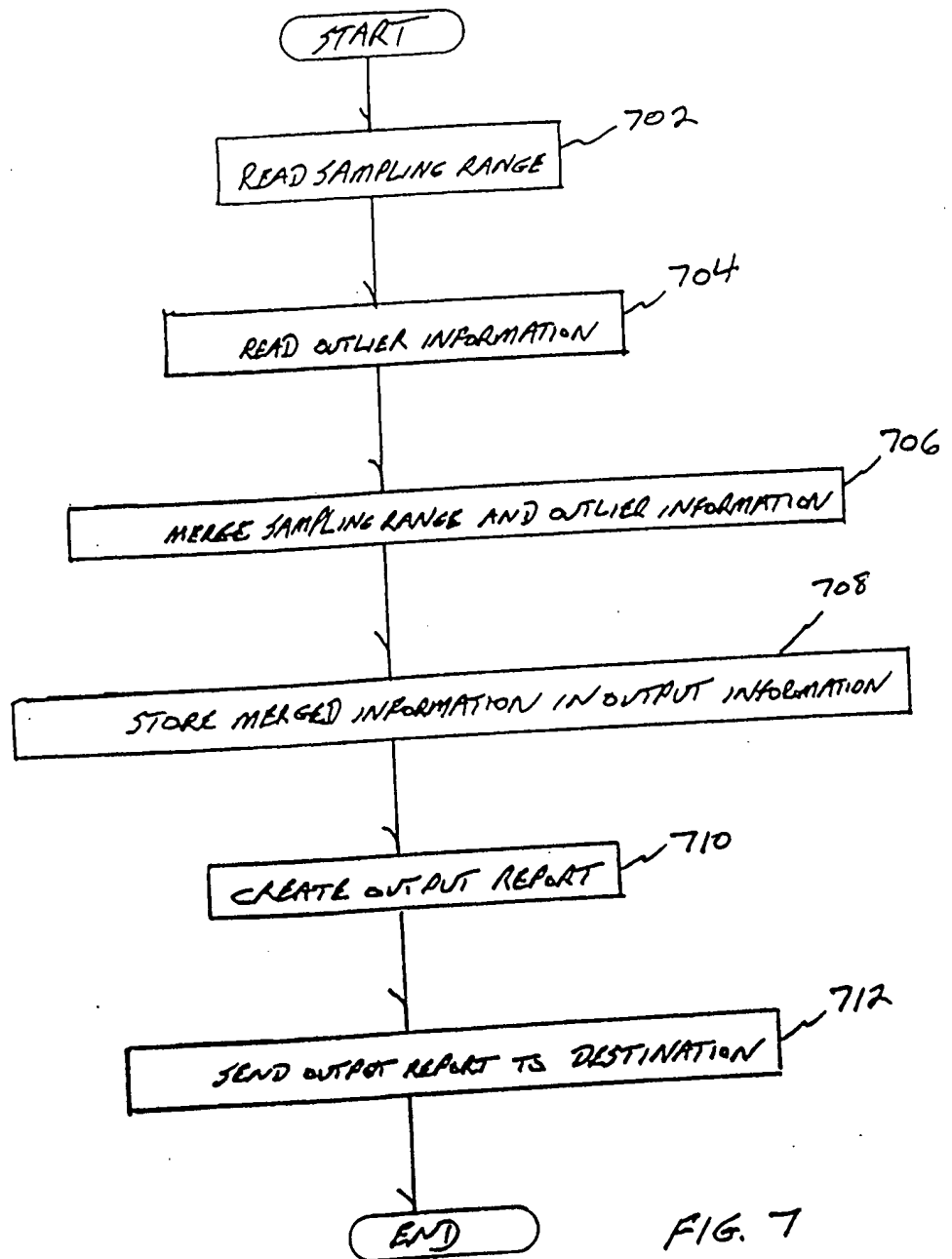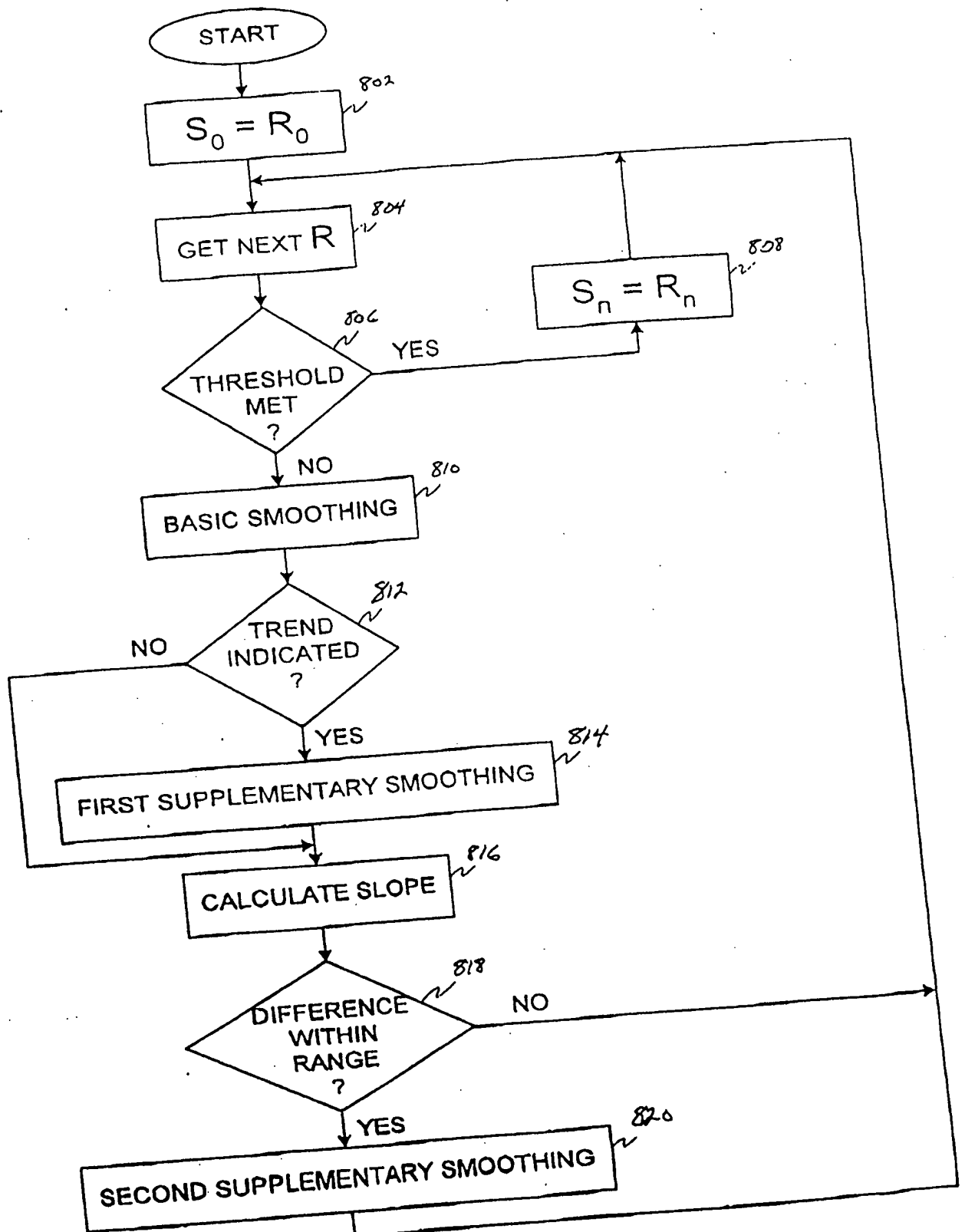
FIG. 7

FIG. 8

Parametric Trend Chart: Results for one test, for an entire wafer
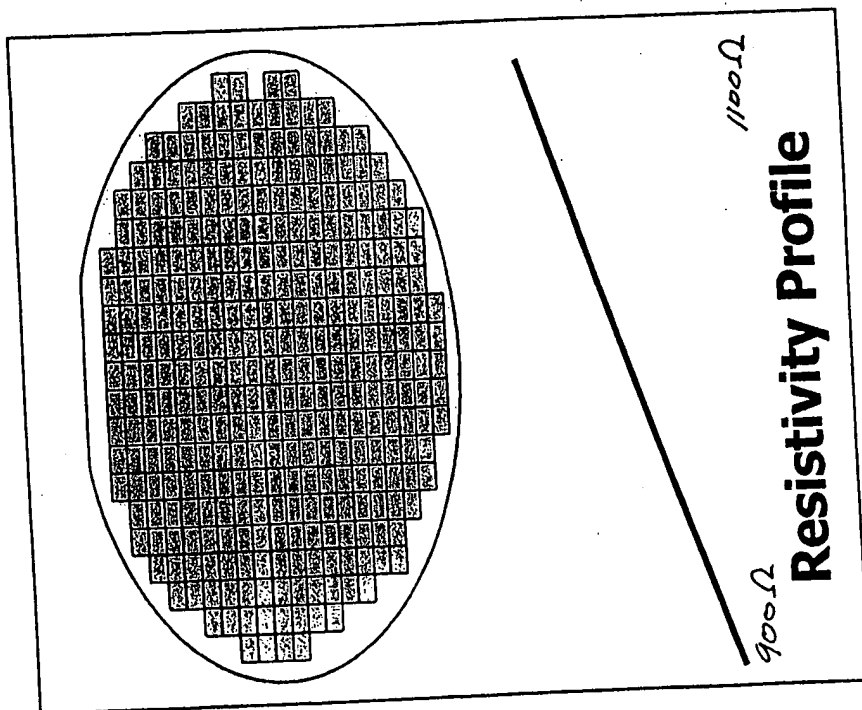

● = Test Limit Outlier
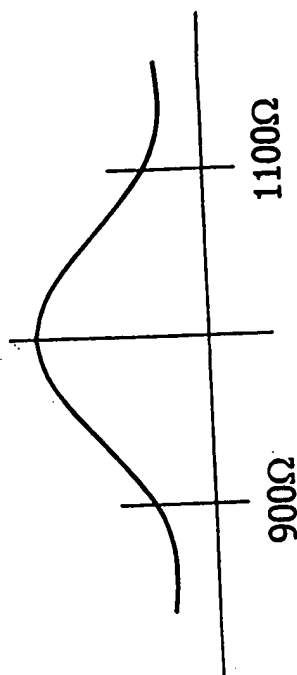
△ = Bin 1 Outlier

FIG. 1

FIG. 11

Resistivity Profile

900Ω   1100Ω

FIG. 10

FIG. 12A

FIG. 12B

START

RECEIVE DATA — 1310

GENERATE COMPOSITE DATA — 1312

DETERMINE EXCLUSION ZONES — 1314

PROXIMITY WEIGHTING — 1316

CLUSTER DETECTION — 1318

SPATIAL ANALYSIS 1320

MERGE DATA — 1322

OUTPUT REPORT — 1324

END
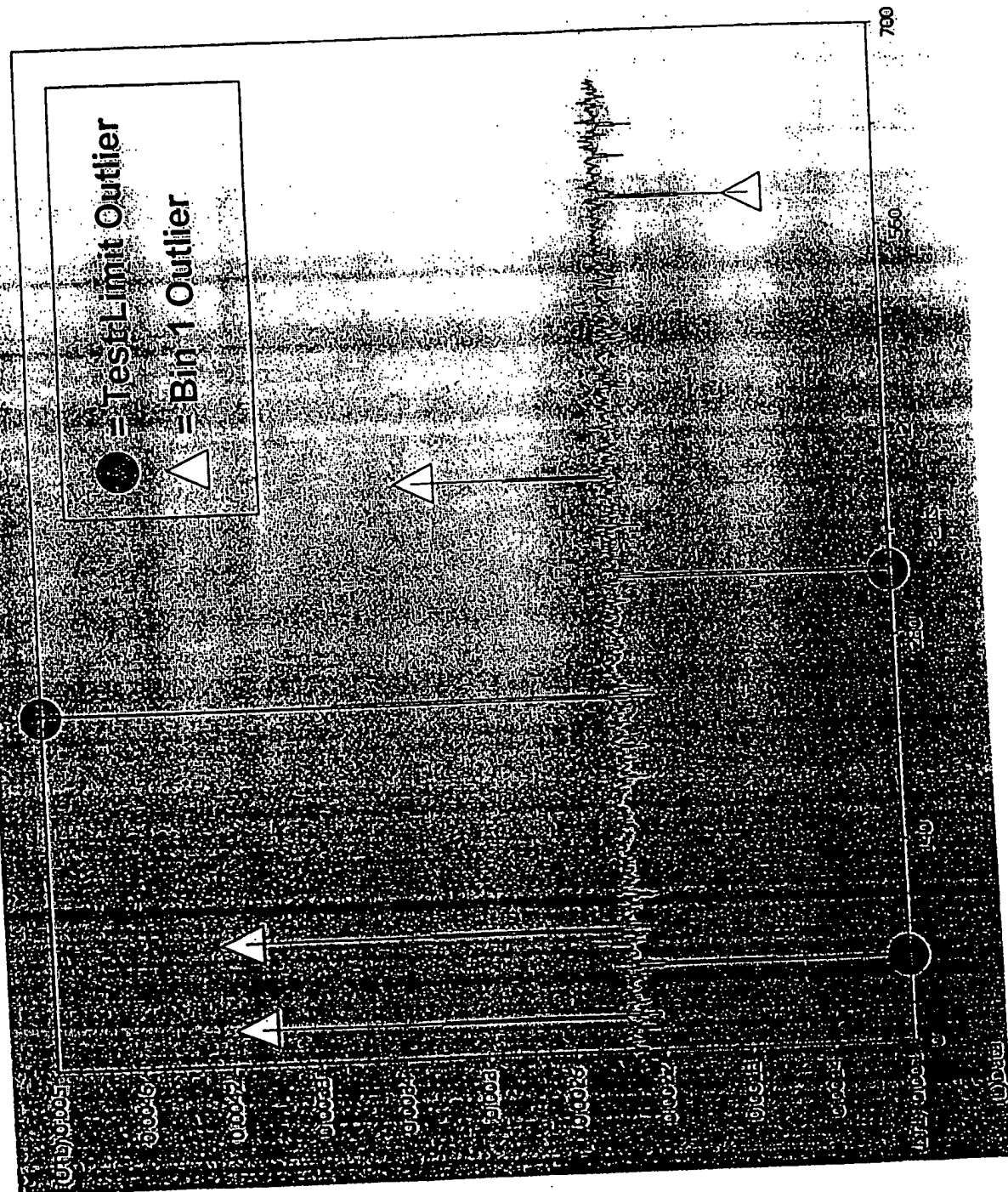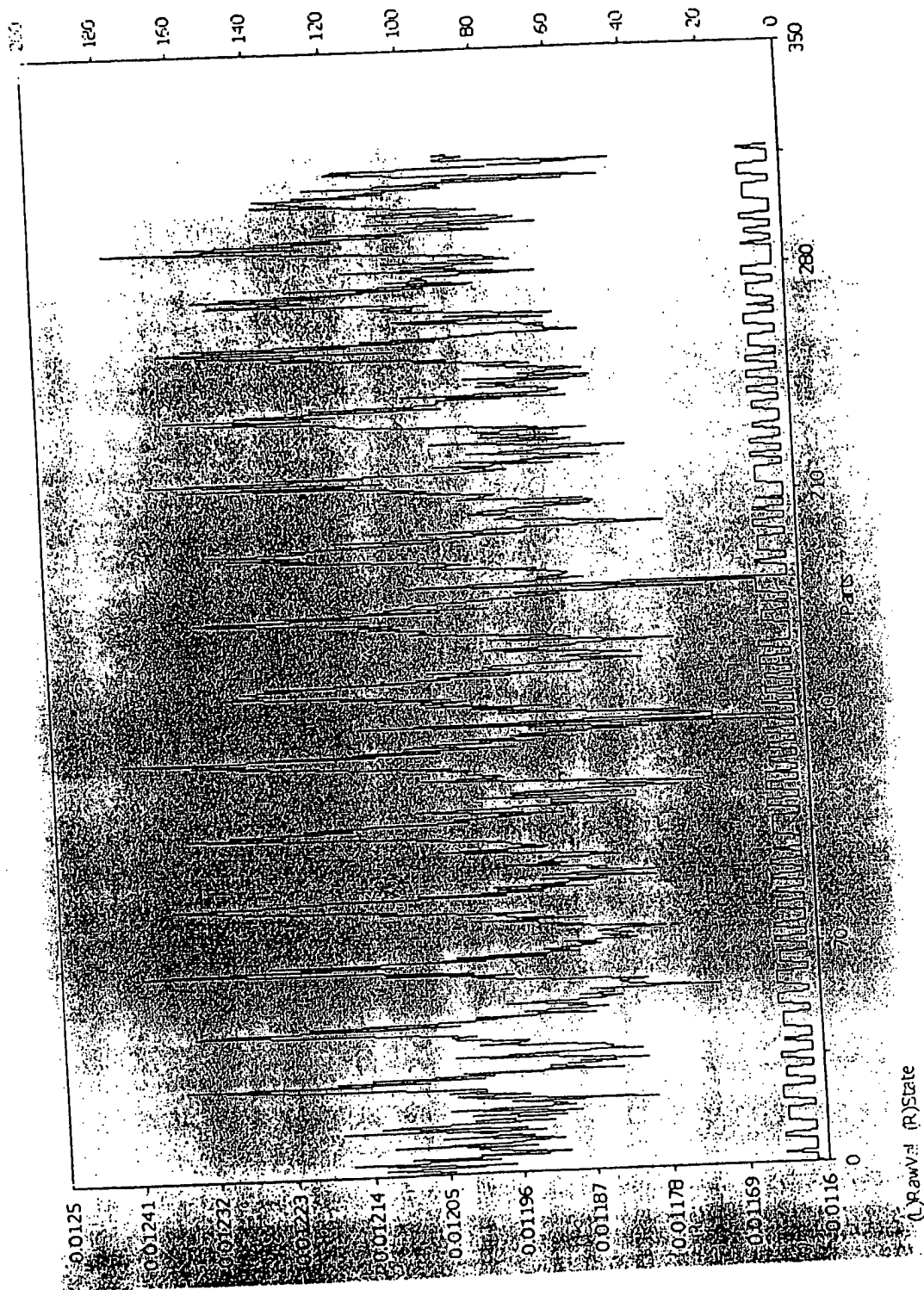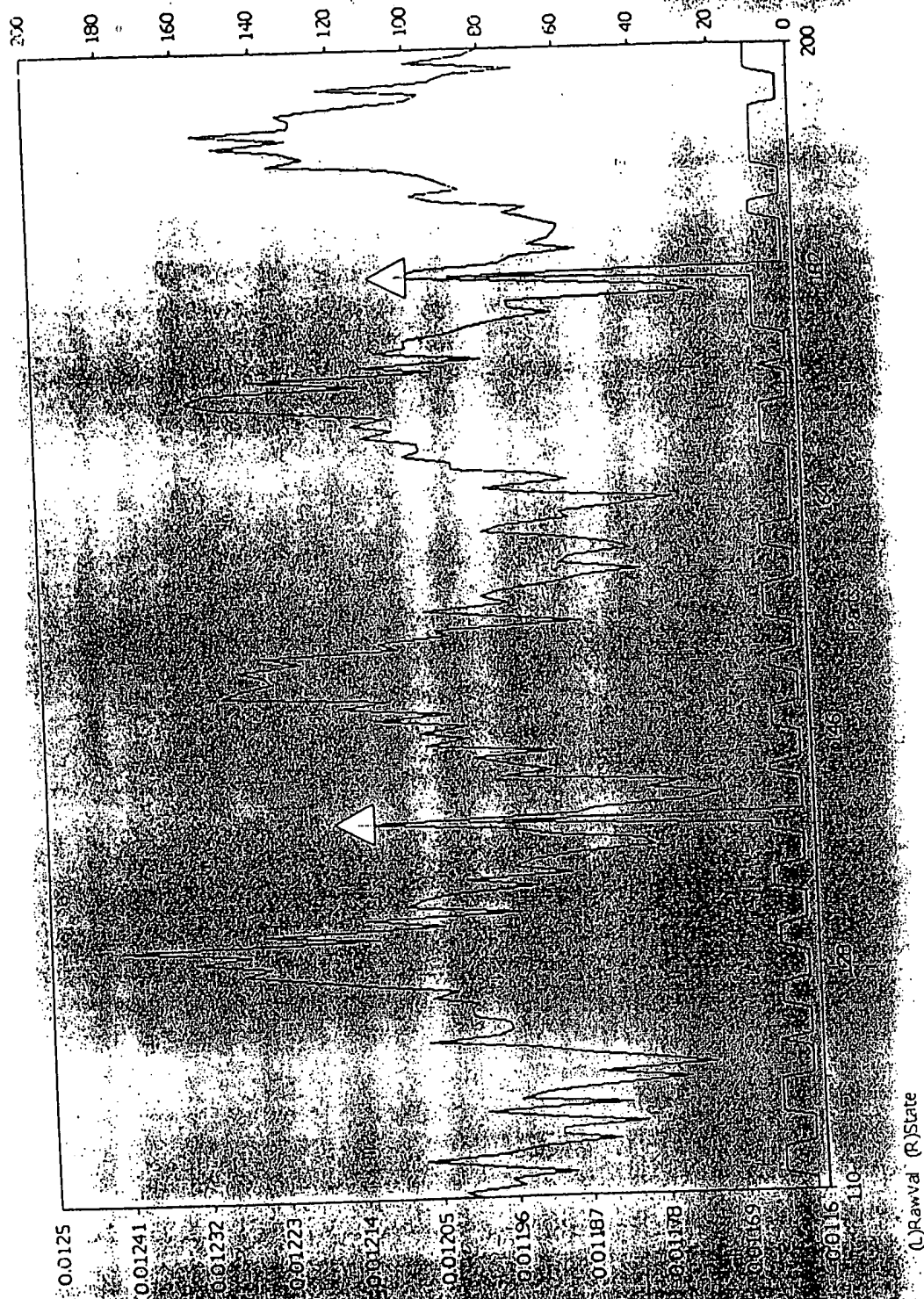
FIG. 13

Composite Addition

FIG. 14

Wafer 03
Wafer 02
Wafer 01
Location 10,12

Legend

⊠ Devices with value of 5

▤ Devices with value of 10

Example:

For the location 10,12 a variable with the definition of five would have a value of two and a variable with a definition of ten wold have a value of one.

Now, assume that the user has a formula of X > 1 and also assume that x is a variable with a definition value of five. According ot the data, x would have a value of 2 (as we saw above) and the formula would equate as true. Since the formula resulted in a true value, it would be included in the composite map.
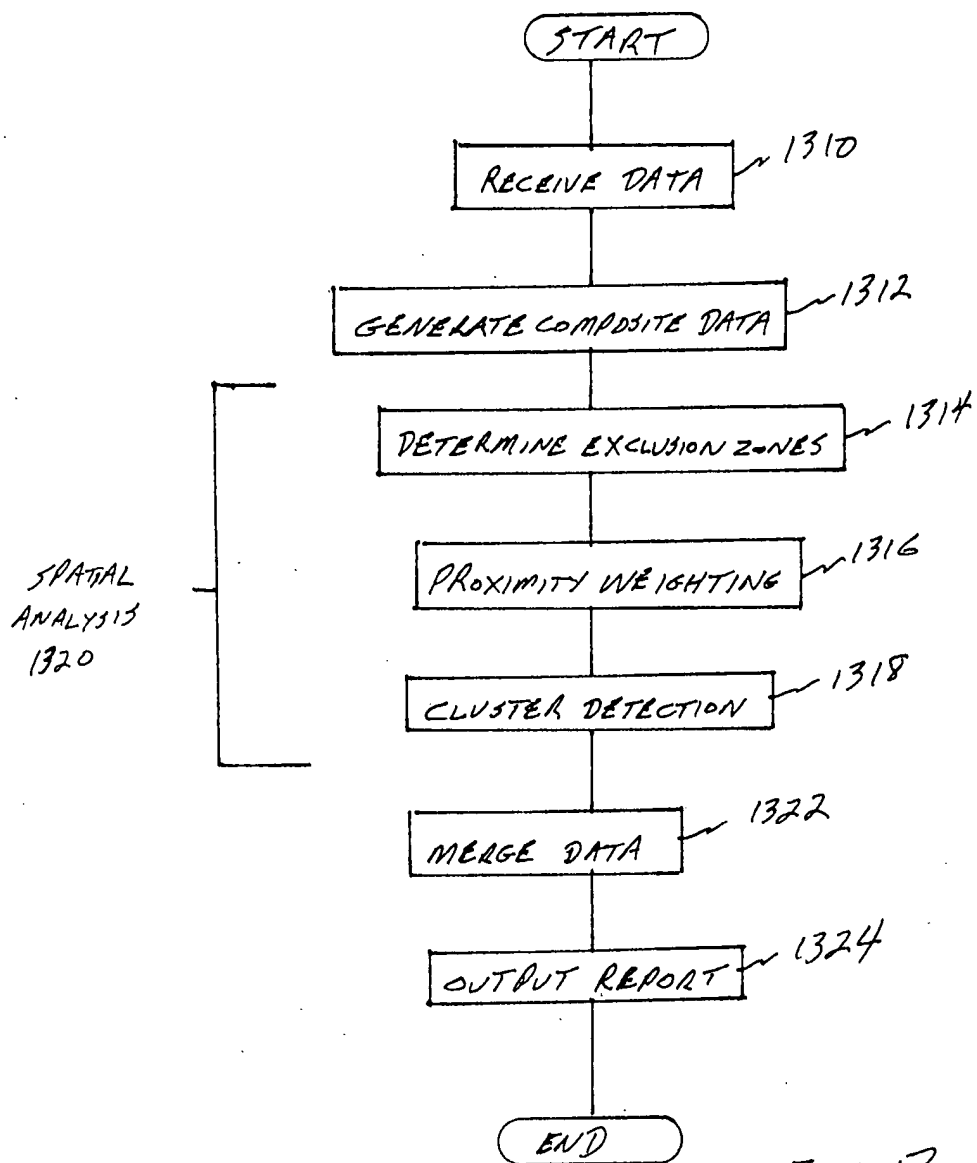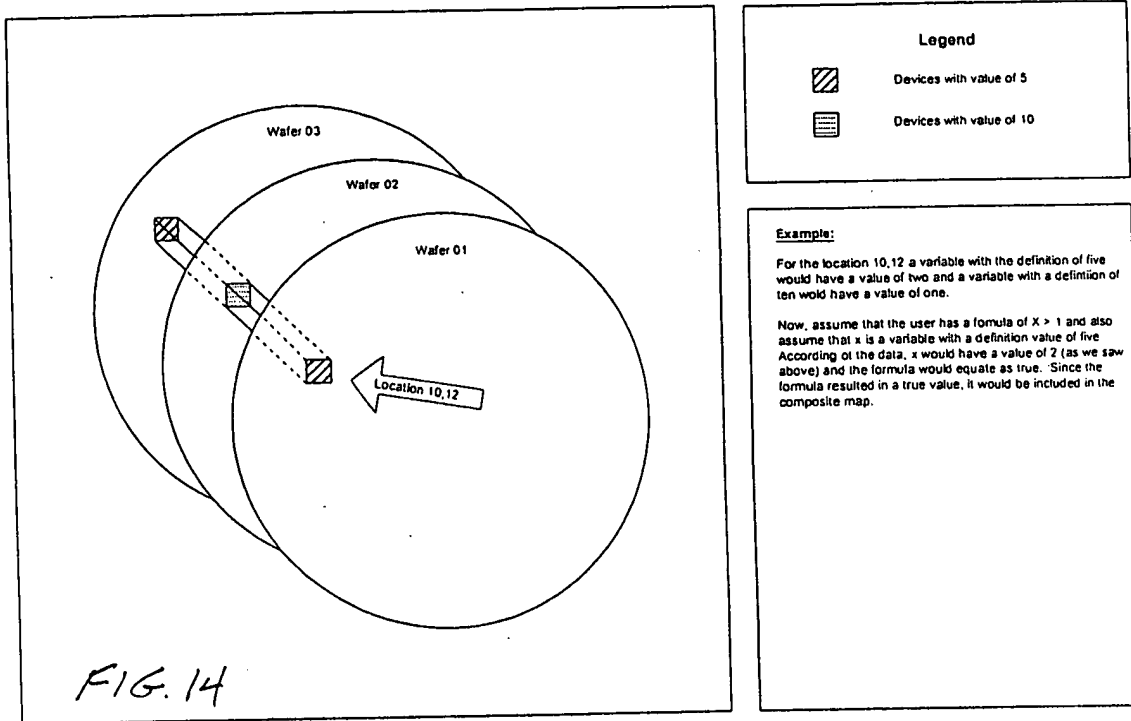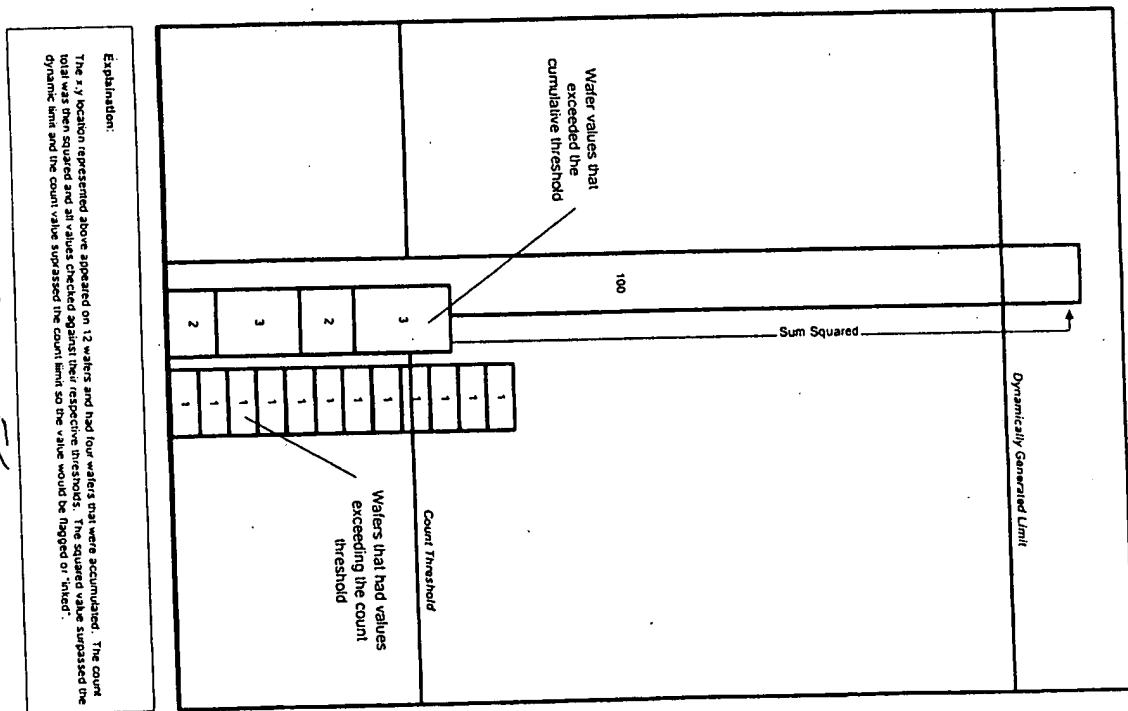

Cumulative Squared Method

Done on X,Y location basis for entire dataset / Lot

FIG. 15C

Wafer values that exceeded the cumulative threshold

100

Sum Squared

Dynamically Generated Limit

Count Threshold

Wafers that had values exceeding the count threshold

Explanation:

The x,y location represented above appeared on 12 wafers and had four wafers that were accumulated. The count total was then squared and all values checked against their respective thresholds. The squared value surpassed the dynamic limit and the count value surpassed the count limit so the value would be flagged or 'inked'.

START

SELECT DATA SET — 1510

SELECT DATA POINT — 1512

VALUE > THRESHOLD ? — 1514
  Y → COUNT = COUNT + 1 — 1516
  N ↓

VALUE > CUMUL VALUE THRSH ? — 1518
  Y → CUMUL = CUMUL + VALUE — 1520
  N ↓

LAST DATA POINT ? — 1522
  N →
  Y ↓

LAST DATASET ? — 1524
  N →
  Y ↓

A

FIG. 15A

A

CALCULATE LIMIT — 1526

SELECT DATA POINT — 1528

SQUARE CUMUL VALUE — 1530

SQUARED VALUES > LIMIT ? — 1532
Y → DESIGNATE FOR COMPOSITE DATA — 1534
N

COUNT > THRSH ? — 1536
Y → DESIGNATE FOR COMPOSITE DATA — 1538
N

LAST DATA POINT ? — 1540
N
Y

END

FIG. 15B

Exclusion Zone Graphical Representation

X-Axis

Y-Axis

Point listed in
exclusion file

Points generated
from exclusion
band

Legend

= Devices Considered for Exclusion

= Devices that met defined criteria, Good_Bins = 0
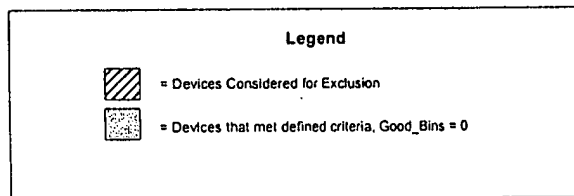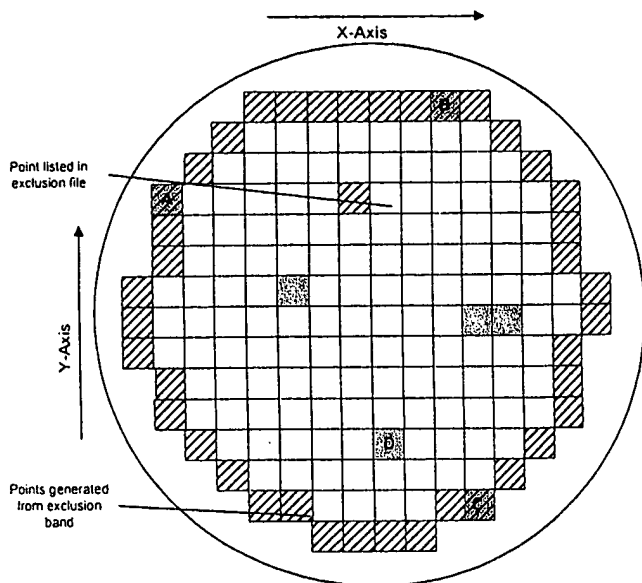
Description

The exclusion zone function is designed exclude
certain devices based on specific x,y locations read
from a file or devices within a certain distance of the
edge of the dataset that meet a specified criteria.
Devices are excluded by setting them to a value which
is out of the range of subsequent processes such as
proximity weighting.

Logic

Loop through each x,y location
If x,y location is within EXCLUSION_RANGE of the
edge of the data set then
Check is x,y location value (bin) meets
EXCLUSION_CRITERIA
If EXCLUSION_CRITERIA is met, set x,y location to
EXCLUSION_INK_VALUE

Process EXCLUSION_FILE and set all x,y locations
listed in file to EXCLUSION_INK_VALUE

Examples

Device A would be excluded
Device B would be excluded
Device C would be excluded
Device D would be included

Parameters

EXCLUSION_RANGE = 1
EXCLUSION_INK_VALUE = -1
EXCLUSION_CRITERIA = "Good_Bins = 0"
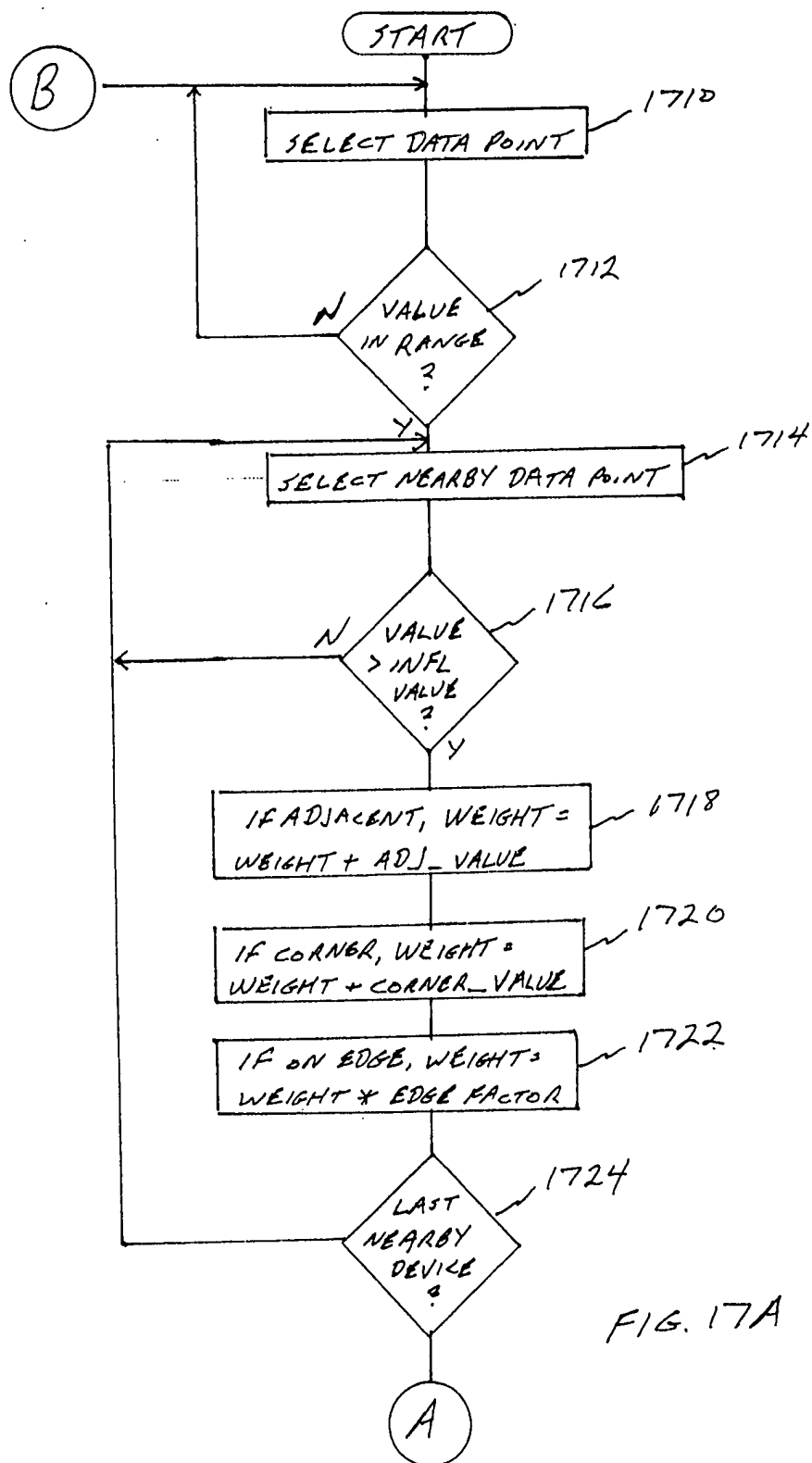EXCLUSION_FILE = "D:/Step2/my.exclude"

FIG. 16

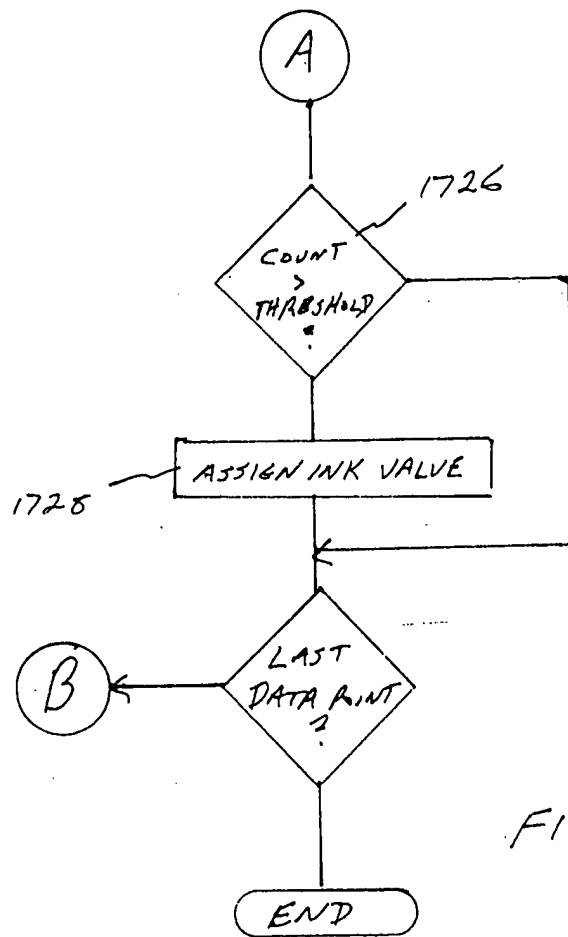START

B

SELECT DATA POINT   1710

N VALUE IN RANGE ?   1712

Y

SELECT NEARBY DATA POINT   1714

N VALUE > INFL VALUE ?   1716

Y

IF ADJACENT, WEIGHT = WEIGHT + ADJ_VALUE   1718

IF CORNER, WEIGHT = WEIGHT + CORNER_VALUE   1720

IF ON EDGE, WEIGHT = WEIGHT * EDGE FACTOR   1722

LAST NEARBY DEVICE ?   1724

A

FIG. 17A

A

1726

COUNT > THRESHOLD

ASSIGN INK VALVE

1728

LAST DATA POINT ?

B

END

FIG. 17B

Proximity Weighting Graphical

Search Pattern

Legend

Z = 30

☐ = 1

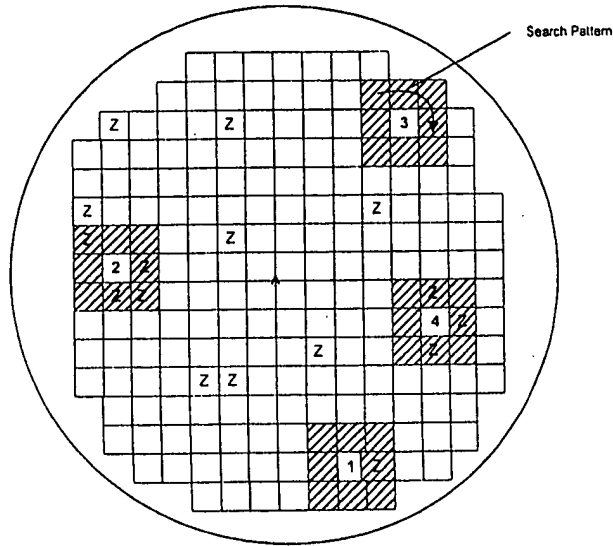▨ = Search Range defined by PROX_RANGE
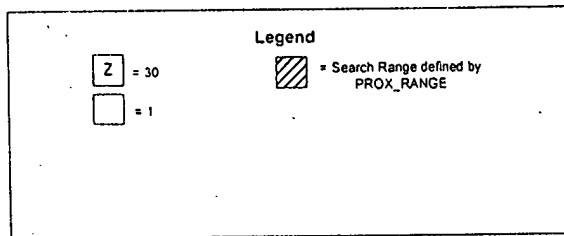
FIG. 18

Description

Proximity Weighting reviews all data points in a data set arranged in x,y fashion and analyzes the points that have values within a specified range. Analysis consists of searching an area, radius of specifed units, around each device and looking for devices that are above a certain threshold. If a device is found that has a value above a said threshold, then it either lends a specified wieght to the device being analyzed; weight depends if the device is adjacent or diagonal to the device being analyzed.

Logic
Loop through each x,y location
If x,y location has value between UPPER_BOUND and LOWER_BOUND
Check all devices in the specified search range (see legend)
If device in search range being checked has a value that meets or exceeds the INFLUENCE_VALUE then
If device is adjacent, increment x,y location count by ADJACENT_VALUE else increment x,y location count by CORNER_VALUE
If x,y location is on edge of dataset then multiply by x,y location count by EDGE_FACTOR
if x,y location count exceeds THRESHOLD, assign x,y location a value of INK_VALUE

Examples
1. Count = 2, end result = 1
2. Count = 6, end result = 20
3. Count = 0, end result = 1
4. Count = 6, end result = 20

Parameters
THRESHOLD = 4 ADJACENT_VALUE = 2
CORNER_VALUE = 1 EDGE_FACTOR = 1
PROX_RANGE = 1 UPPER_BOUND = 30
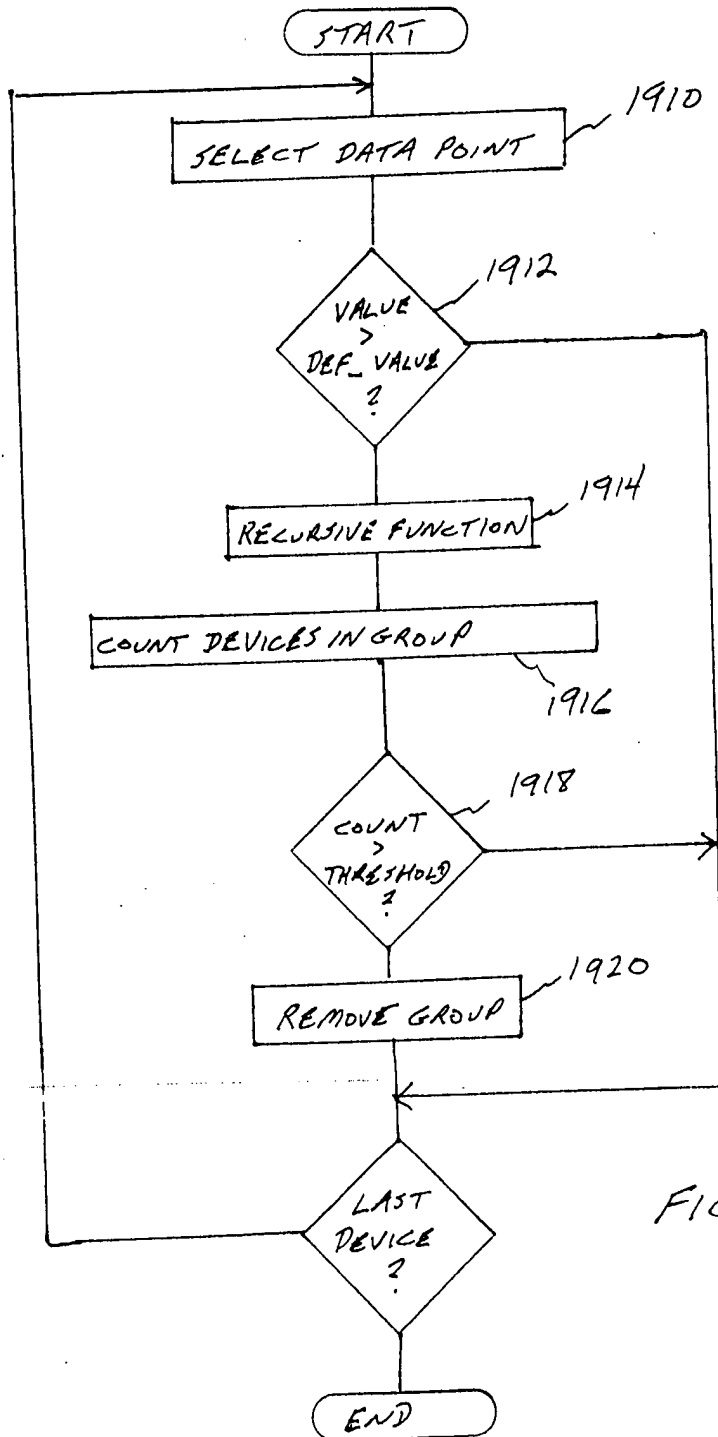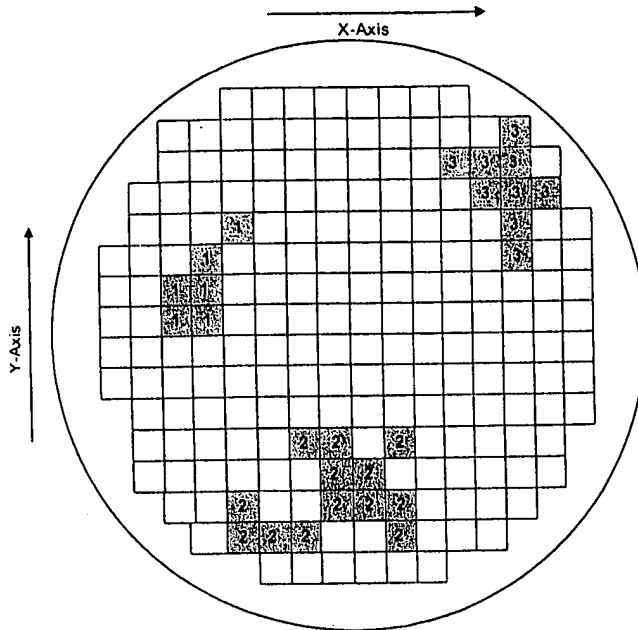LOWER_BOUND = 0 INK_VALUE = 20
INFLUENCE_VALUE = 30

START

SELECT DATA POINT — 1910

VALUE > DEF_VALUE ? — 1912

RECURSIVE FUNCTION — 1914

COUNT DEVICES IN GROUP — 1916

COUNT > THRESHOLD ? — 1918

REMOVE GROUP — 1920

LAST DEVICE ?

END

FIG. 19

Cluster Detection

X-Axis →

Y-Axis ↑

Legend

[shaded box] = Devices that have values matching the DEFINITION_VALUE

[n] = Group Index

Description

The cluster detection function's purpose is to remove goups of locations (defined by the DEFINITION_VALUE) in datasets by assertaining how large each group is and if smaller than a said threshold, remove the group(s).

Logic

Loop through each x,y location
If there is a device at the x,y location
If x,y location has value equal to DEFINITION_VALUE
Call recursive function find next device in group and assign group index

Once group indexes have been assigned, count how many devices are assigned to each group
If group count is less than SIZE_THRESHOLD, remove it / set value to bin one

Examples

Group 1: Would be removed
Group 2: Would be kept
Group 3: Would be kept

Parameters

SIZE_THRESHOLD = 8
DEFINITION_VALUE= 30
INCLUDE_CORNERS = 1

FIG. 20

Legend

Tester Bin Failures in original data

Failure Patter in Composite Map 1

Failure Patter in Composite Map 2

Composite Maps Overlap

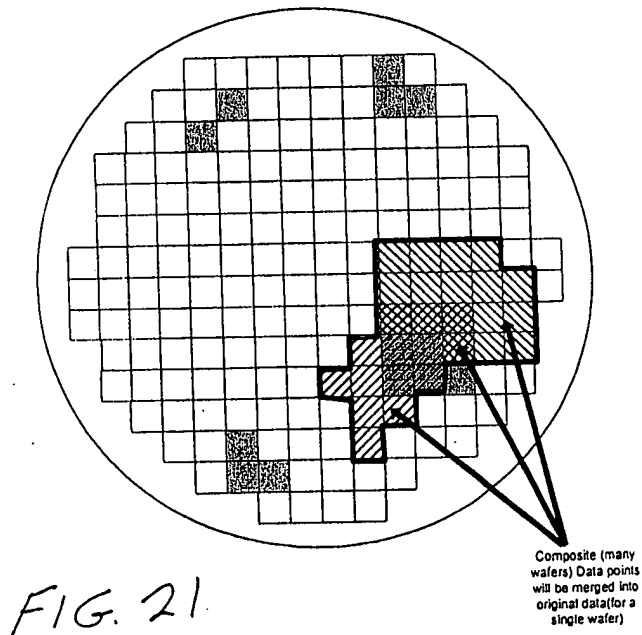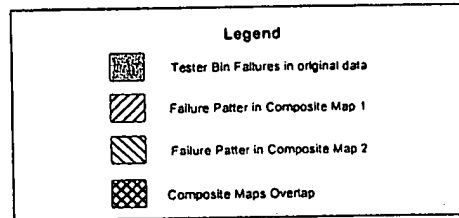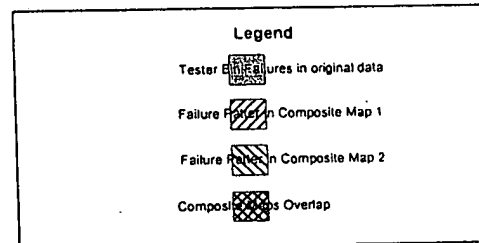**Description:**

Looking at the example to the left, notice that there are two composite map patters, labeled one and two. Notice that there is an area where the two patterns overlap. It is this area and only this area of overlap that will be merged into the original data if and only if a user defined criteria is met. This criteria could be a fixed threshold, 15 for instance or a calculated value such as 90% of the data points constituting the patterns must overlap. In either case, the overlap on the upper left portion of the diagram would have sufficent overlap and would be merged but the overlap on the right of the diagram has only four devices and would not be merged.

Note: It is also possible that instead of merging only the overlapping patterns that all patterns are merged regardless of the criteria and the data points that do meet the criteria are only noted in an output file.

Patterns have enough data points for a 90% criteria match and overlapping data points would be merged into original data

Pattern does not have enough overlap to meet a 90% criteria and would not be merged into the original data

FIG. 24

Absolute Composite Merge

Legend

Tester Bin Failures in original data

Failure Patter in Composite Map 1

Failure Patter in Composite Map 2

Composite Maps Overlap

Description:

Looking at the example to the left, notice that there are two composite map patters, labeled one and two., any points from either pattern are merged in this type of merge operation.

Composite (many wafers) Data points will be merged into original data(for a single wafer)

FIG. 21

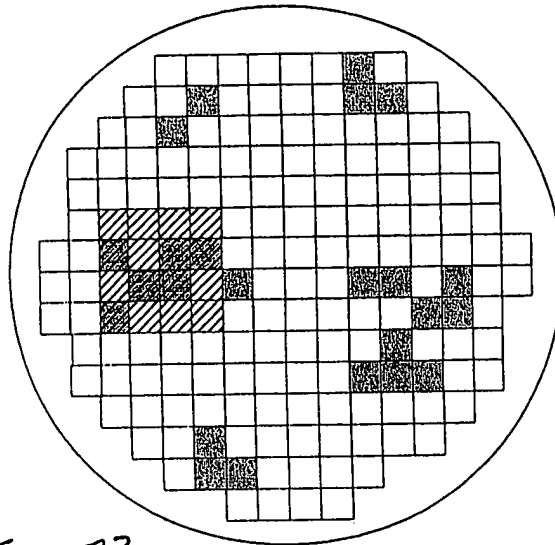Composite Pattern Constitution

FIG. 23

Legend

Tester Bin Failures in original data

Failure Pattern in Composite Map

Pattern Constitution Mode

Given the example on the top left and assuming that the user has set the criteria for a pattern merge is that at least 50% of the composite map pattern must match the original data we can see that the composite pattern would not be merged on the originall data. This is due to the failure of the data to pass the criteria specified. at least 8 out of the 16 data point pattern must have corresponding point on the original data, of which there are only six. Since the pattern failed the criteria it will not be merged with the original data.

Absolute Mode

In this mode, the example to the left would have the composite map super imposed over the original data map regardless of how much of the pattern is representitive of the actual data.

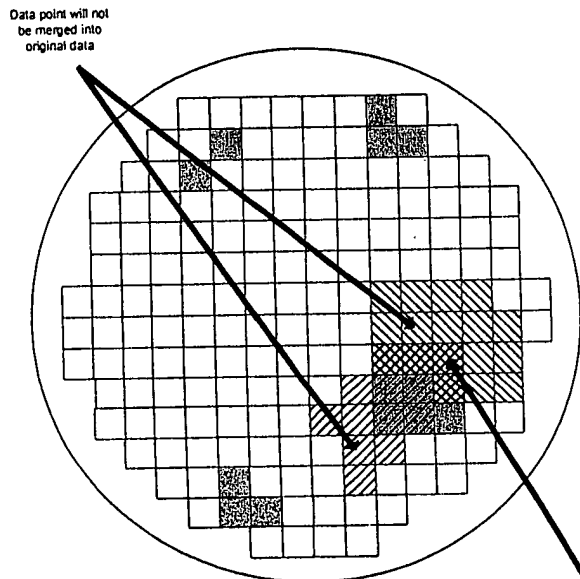Multiple Recipe / Overlapping Composite Map Definition

Data point will not
be merged into
original data

FIG. 22

Data point will be
merged into
original data

Legend

Tester Bin Failures in original data

Failure Patter in Composite Map 1

Failure Patter in Composite Map 2

Composite Maps Overlap

Description:

Looking at the example to the left. notice that there are two composite map patters. labeled one and two. Notice that there is an area where the two patterns overlap. It is this area and only this area of overlap that will be merged into the original data. the rest of the pattern is descarded.
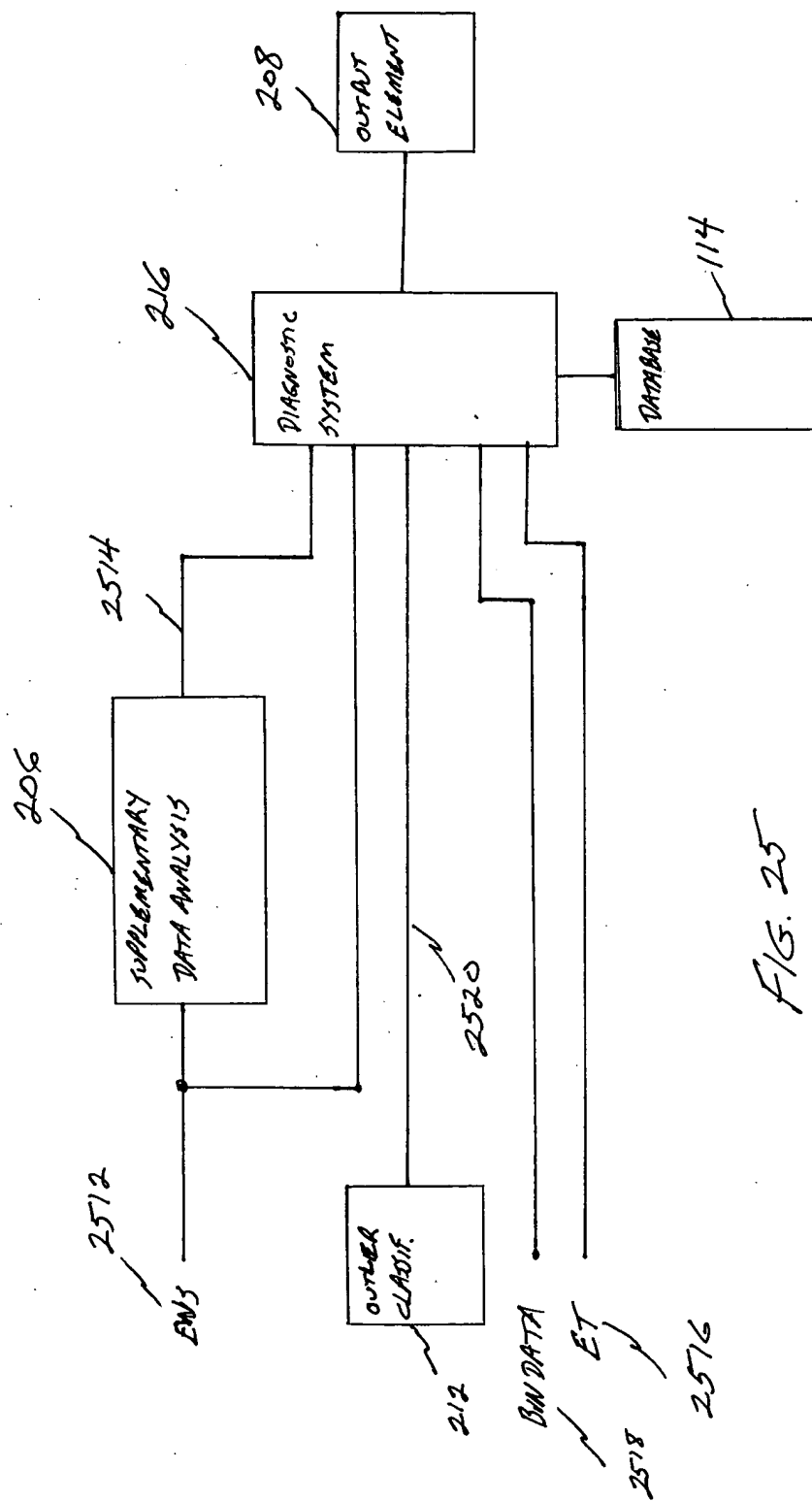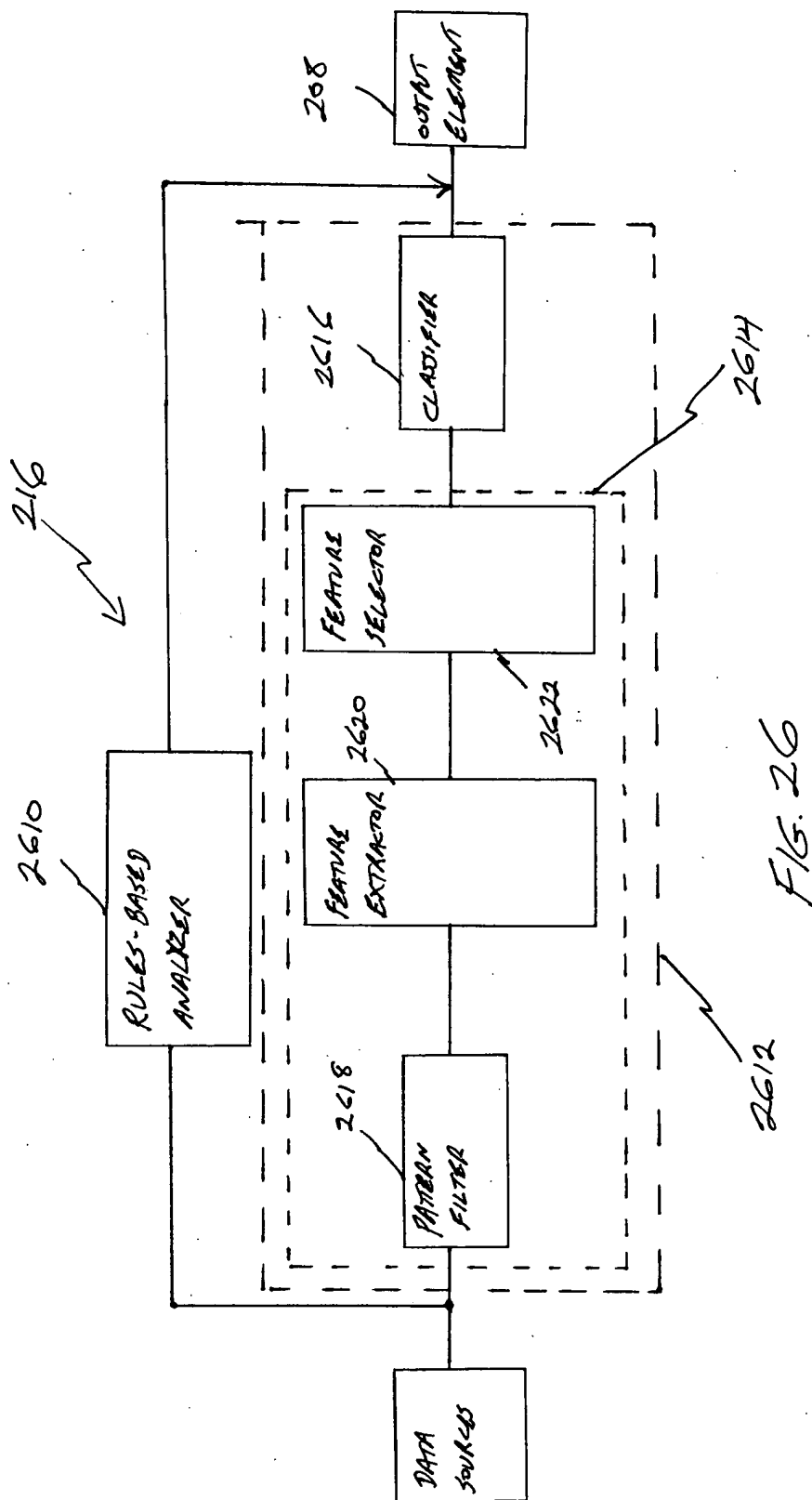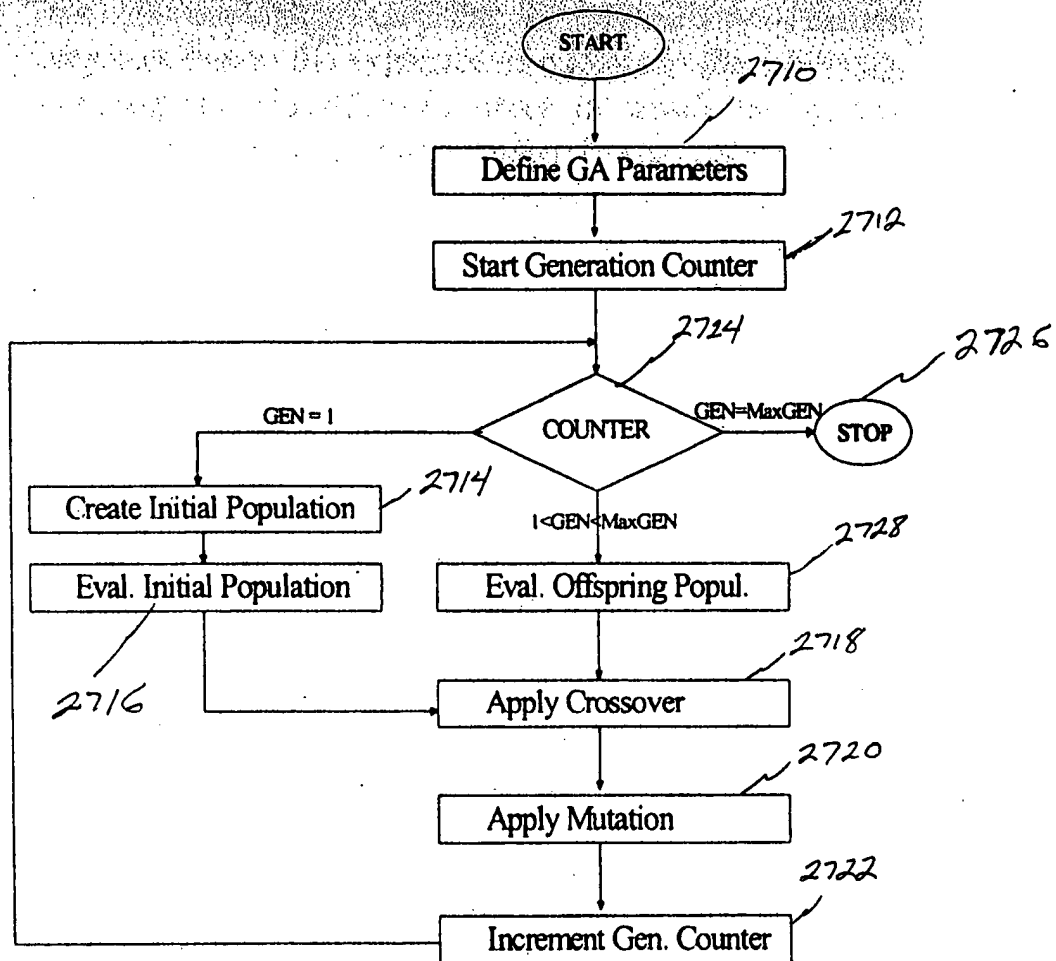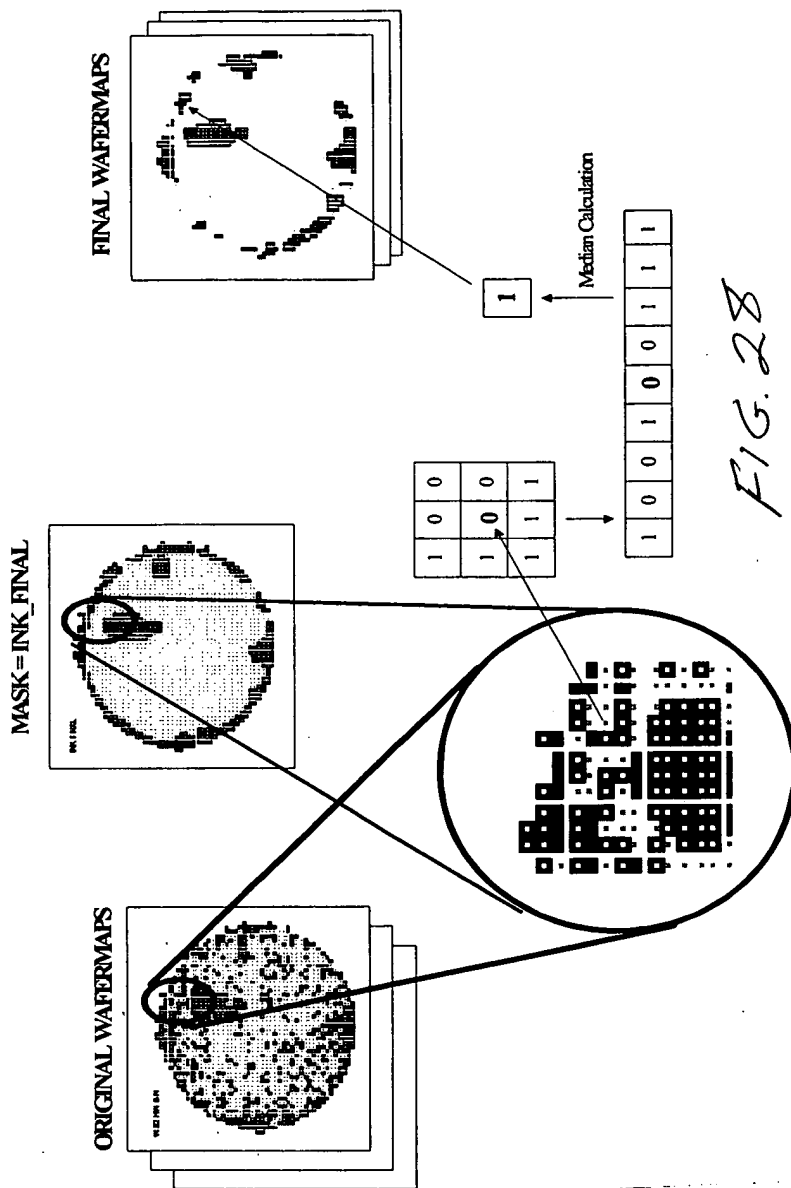
208

OUTPUT
ELEMENT

216

DIAGNOSTIC
SYSTEM

114

DATABASE

2514

206

SUPPLEMENTARY
DATA ANALYSIS

2520

2512

EMS

212

OUTLIER
CLASSIF.

BIN DATA

2518

ET

2516

FIG. 25

FIG. 26

START

2710

Define GA Parameters

Start Generation Counter

2712

2724

COUNTER

2726

STOP

GEN=MaxGEN

GEN = 1

Create Initial Population

2714

1<GEN<MaxGEN

Eval. Offspring Popul.

2728

Eval. Initial Population

2716

Apply Crossover

2718

Apply Mutation

2720

Increment Gen. Counter

2722

FIG. 27

FINAL WAFERMAPS

MASK = INK_FINAL

ORIGINAL WAFERMAPS

Median Calculation

FIG. 28

Input Layer
2912

Hidden Layer 2914

Output Layer
2916

FIG. 29